НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
Кафедра інформаційної безпеки

«На правах рукопису»

УДК 004.056

«До захисту допущено»
В.о. завідувача кафедри
_____ М.В.Грайворонський
“___”_____2018 р.

Магістерська дисертація

на здобуття ступеня магістра

зі спеціальності: 125 Кібербезпека

на тему: Модель корпоративної мережі підприємства з використанням системи IDS/IPS

Виконав (-ла): студент (-ка) 2 курсу, групи ФБ-71мп
(шифр групи)

Коваленко Андрій Олегович
(прізвище, ім’я, по батькові)

Науковий керівник к.е.н., доц. Ткач Володимир Миколайович __ _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант _____ _____ _____
(назва розділу) (науковий ступінь, вчене звання, , прізвище, ініціали) (підпис)

Рецензент зав. кафедрою ПС КНУ Лєбєдєва Є.О.
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студент _____
(підпис)

Київ – 2018 року

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
Кафедра інформаційної безпеки

Рівень вищої освіти – другий (магістерський) за освітньо-професійною програмою
Спеціальність (спеціалізація) – 125 Кібербезпека («Системи і технології кібербезпеки)

ЗАТВЕРДЖУЮ
В.о. завідувача кафедри
_____ М.В.Грайворонський
(підпис)

«___»_____2018 р.

ЗАВДАННЯ
на магістерську дисертацію студенту

Коваленко Андрію Олеговичу

1. Тема дисертації: Модель корпоративної мережі підприємства з використанням системи IDS/IPS
науковий керівник дисертації к.е.н., доц. Ткач Володимир Миколайович,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «15» листопада 2018 р. № 4171-с

2. Термін подання студентом дисертації 12.12.2018 р.

3. Об'єкт дослідження _____
_____

4. Вихідні дані _____
_____
_____
_____

5. Перелік завдань, які потрібно розробити _____
_____
_____
_____
_____
_____

6. Орієнтовний перелік ілюстративного матеріалу _____
_____
_____

7. Орієнтовний перелік публікацій _____
_____

8. Консультанти розділів дисертації*

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

9. Дата видачі завдання _____

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка

Студент _____ (підпис) _____ (ініціали, прізвище)

Науковий керівник дисертації _____ (підпис) _____ (ініціали, прізвище)

* Консультантом не може бути зазначено наукового керівника магістерської дисертації.

РЕФЕРАТ

Робота складається з 87 сторінок, і містить 10 ілюстрацій, 23 таблиці та 6 джерел.

Метою роботи є розробка системи автоматизованого збору та перевірки мережевих налаштувань.

Результатом роботи є програмний засіб, що базується на розробленій політиці безпеки, та проводить автоматизовано перевірку налаштувань мережного обладнання.

Ключові слова: МЕРЕЖА, СПИСОК ДОСТУПУ, ПОЛІТИКА БЕЗПЕКИ, ПРАВИЛО, ВРАЗЛИВІСТЬ

ABSTRACT

The work consists of 87 pages, and contains 10 illustrations, 23 tables and 6 sources.

The aim of the work is to develop a system for automated collection and validation of network settings.

The result of the work is a software tool based on the developed security policy, and conduct automated testing of network equipment settings.

Keywords: NETWORK, ACCESS LIST, SAFETY POLICY, RULES, VARIABILITY

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ПЗ – Програмне забезпечення;

ІБ – Інформаційна Безпека;

ПБ – Політика Безпеки;

СД – Список доступу

ВСТУП

Маршрутизатори були розроблені для передачі дейтаграм до свого місця призначення і були програмовані схемамами рішення проблем, таких як затори або збій сегмента мережі. Ці схеми включають в себе зміну маршруту дейтаграм на альтернативний напрям. Тому неможливо стверджувати з будь-якою точністю - який шлях буде обраний дейтаграмою, мандруючою за межі локальної мережі.

Дейтаграмма може рухатися по прямому маршруту, або, швидше за все, подорожувати через кілька маршрутизаторів, розміщених в будь-яких куточках світу. Ці маршрутизатори, найімовірніше, не належать відправнику або одержувачу, а третій стороні. У більшості випадків це не має значення, однак дейтаграми можуть бути скопійовані, і їх безпека може піддаватися ризику, коли вони подорожують через маршрутизатор, без повідомлення відправника або одержувача.

Процес відомий як моніторинг пакетів, має багато законних цілей, включаючи аналіз функціонування мережі та дотримання законності, однак при цьому програми для моніторингу мережі зараз доступні будь-кому, хто вирішить їх використовувати. У минулому, моніторинг пакетів вимагав комп'ютера, підключеного кабелем до мережі.

Об’єктом дослідження є мережеве обладнання CISCO.

Предмет дослідження – власне налаштування обладнання.

1 ТЕХНОЛОГІЇ, ЩО ВИКОРИСТОВУЮТЬСЯ В ДИПЛОМІ

В даному розділі будуть визначенні та описані базові для роботи поняття: мережі, мережеві топології, види мережевого обладнання. Будуть описані особливості та принципи функціонування мережевого обладнання.

1.1 Мережі та їх види

Комп'ютерна мережа (Computer Network) - це безліч комп'ютерів, з'єднаних лініями зв'язку, що працюють під управлінням спеціального програмного забезпечення.

Під лінією зв'язку зазвичай розуміють сукупність технічних пристроїв, і фізичного середовища, що забезпечують передачу сигналів від передавача до приймача. У реальному житті прикладами ліній зв'язку можуть служити ділянки кабелю і підсилювачі, що забезпечують передачу сигналів між комутаторами. На основі ліній зв'язку будуються канали зв'язку.

Каналом зв'язку зазвичай називають систему технічних пристроїв і ліній зв'язку, що забезпечує передачу інформації між абонентами. Співвідношення між поняттями "канал" і "лінія" описується наступним чином: канал зв'язку може включати в себе кілька різнорідних ліній зв'язку, а одна лінія зв'язку може використовуватися декількома каналам.

1.1.1 Типи мереж

Персональна мережа (Personal Area Network, PAN) - дозволяє пристроям обмінюватися даними на невеликих відстанях. PAN об'єднує такі пристрої як миші, клавіатури, принтери, смартфони, планшети, тощо. Найбільш поширеною технологіэю підключення є Bluetooth (технологія отримала назву на честь короля вікінгів Харальда I Синезубого, який об'єднав народи на території сучасних Данії і Сконе).

PAN також може бути створена за допомогою інших технологій, що дозволяють обмінюватися даними на малих відстанях (наприклад, RFID - Radio Frequency IDentification - спосіб автоматичної ідентифікації об'єктів при якому дані, що зберігаються в транспондерах, або RFID-мітках зчитуються за допомогою радіосигналів).

Локальна мережа (Local Area Network, LAN) - це комп'ютерна мережа, яка, як правило, покриває невелику територію, розташовуючись в одному або декількох будівлях.

Термін «локальна» в даному контексті належить до спільної локальному управління (не означає обов'язкову фізичну близькість компонентів один до одного). Локальної може бути домашня мережа, об'єднання комп'ютерів і інших пристроїв малого офісу або великого підприємства.

В LAN широко використовуються дротяні з'єднання, більшість з яких виконується за допомогою мідних проводів, а деякі - оптоволоконних. Зазвичай, провідні мережі працюють на швидкостях від 100 Мбіт / с до 1 Гбіт / с. Більш сучасні LAN можуть працювати зі швидкістю 10 Гбіт / с. Найбільш поширеним стандартом проводового з'єднання є стандарт IEEE 802.3, зазвичай званий Ethernet.

У локальних мережах поряд з провідними технологіями широко використовуються бездротові з'єднання за стандартом IEEE 802.11, більш відомим як Wi-Fi. Бездротові мережі Wi-Fi працюють на швидкостях від декількох до сотнею мегабіт в секунду.

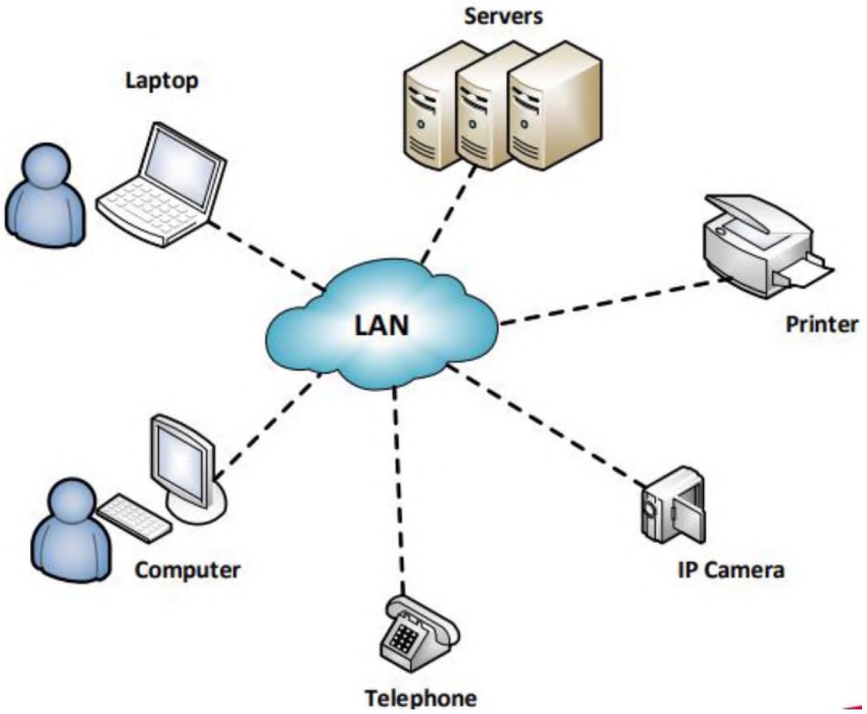


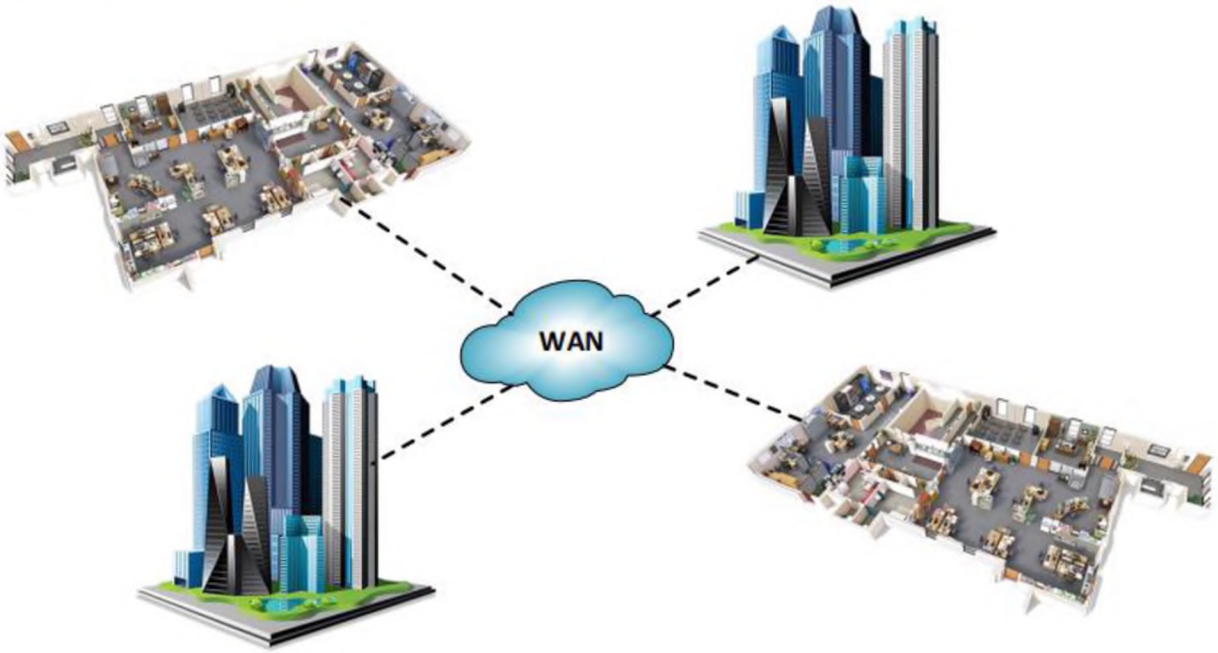
Рисунок 1.1- Локальна мережа

Муніципальні мережі (metropolitan area network, MAN) - об'єднують комп'ютери в межах міста. Як приклад можна розглянути систему кабельного телебачення, в якій, завдяки певним змінам, з'явилася можливість передачі цифрових даних і, з часом, система перетворилася в муніципальну комп'ютерну мережу.

Глобальна мережа (Wide Area Network, WAN) - охоплює значні території, з'єднує локальні мережі, які можуть розташовуватися в географічно віддалених місцях. Глобальна мережа схожа на велику дротову локальну комп'ютерну мережу, але існують важливі відмінності:

- управління локальними мережами та надання доступу до міжмережевий середовища передачі даних здійснюється різними організаціями;
- можуть з'єднуватися мережі, що використовують різні види мережевих технологій;
- за допомогою комунікаційних каналів можуть зв'язуватися окремі комп'ютери з локальними мережами, або цілі мережі.

Рис 1.2 Глобальна мережа



1.1.2 Мережеві топології

Оперативний взаємозв'язок між комп'ютерами по локальній мережі здійснюється за допомогою ліній зв'язку. Вся система, в залежності від фізичного підключення вузлів, а також, самого геометричного розташування вузлів мережі, називають **мережевою топологією**.

Існують логічна і фізична топології, які є незалежними між собою. Фізична топологія здійснює в мережі геометрію побудови, а логічна встановлює в мережі для всіх потоків даних їх напрямок і спосіб передачі.

У локальних мережах найбільш затребувані фізичні топології, такі як:

- «Шина» (bus);
- "Зірка" (star);
- "Кільце" (ring);
- а також, логічне «кільце» (або Token Ring).

Мережа з наявністю **шинної топології**. Тут для передачі даних використовується коаксікальний кабель (моноканал), на кінцях його

встановлюються термінатори, або кінцеву опору. Підключення кожного комп'ютера до кабелю відбувається через Т-роз'єм (Т-коннектор). Через передавальний вузол мережі дані по шині передаються в обидві сторони, при цьому відбиваються від термінаторів. Іншими словами, термінатори гасять сигнали, які досягають до кінця каналів передачі даних. Таким чином, інформація, що передається проходить через всі вузли, але приймається і фіксується тільки одним, якому і призначалася. Логічна шинна топологія забезпечує в мережі спільну і одночасну передачу інформації до всіх ПК, і навпаки, всі дані від ПК в усі напрямки передаються по мережі. Такий вид передачі сигналів називають ще широкомовним.

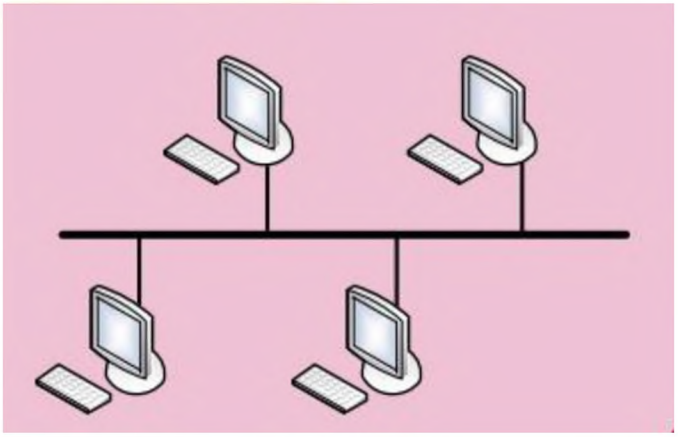
Мережі шинної топології мають і свої переваги:

- легко налаштовується і конфігуруються;
- стійкість даної мережі до окремих неполадок в вузлах;
- якщо один з вузлів виходить з ладу, це ніяк не впливає на працездатність всієї мережі.

Але є й недоліки:

- обмеження в кількості робочих станцій і довжині кабелю;
- може зупинитися вся робота мережі в разі розриву кабелю;
- складно визначати дефекти в з'єднаннях.

Рис. 1.3 Топологія “Шина”



Топологія мережі - **"зірка"**

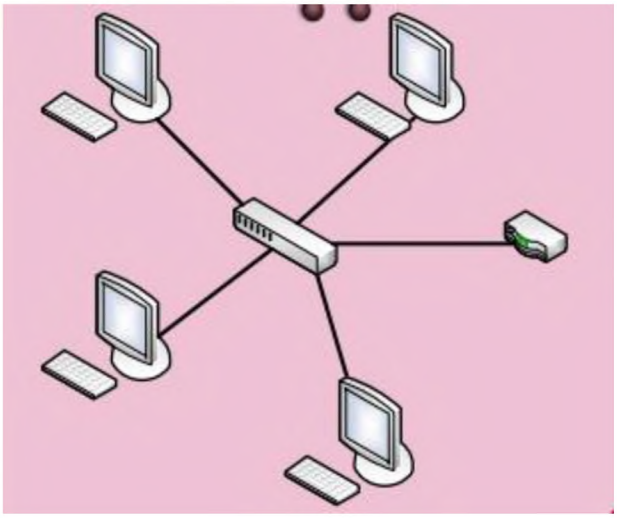
В даній мережі кожна окрема робоча станція кабелем (звита пара) приєднана до хабу або концентратора, що забезпечує для всіх ПК паралельне з'єднання (всі комп'ютери мережі можуть один з одним спілкуватися).

Дані, які відправляються від однієї передавальної станції, через хаб і всі лінії йдуть на усі ПК. Іншими словами, інформація може надходити на будь-яку робочу станцію, але приймати її можуть лише ті станції, яким вона призначена. Оскільки передача сигналів даної типології фізична «зірка» і вона широкомовна, то логічна топологія в такій локальній мережі буде логічної шиною. В основному застосовується для локальних мереж, що мають архітектуру 10Base-TEthernet.

Переваги даної топології зірка:

- легке підключення нового ПК;
- централізоване управління;
- стійкість мереж до несправностей ПК;
- стійкість до розривів в окремих з'єднання ПК.

Рис. 1.4 Топологія “Зірка”



Топологія мережі **"кільце"**

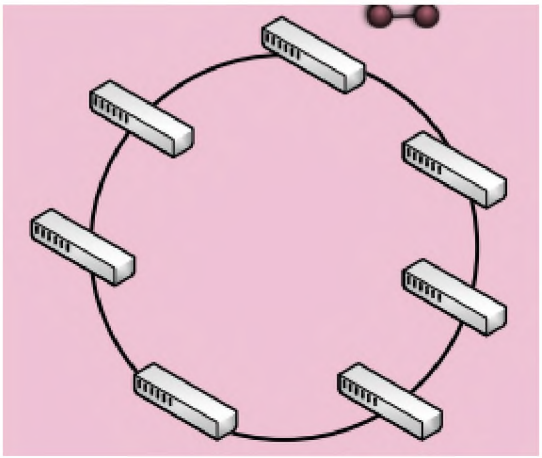
Нерозривне кільце, за допомогою якого передається інформація між ПК, в топології мережі забезпечується з'єднанням усіх вузлів каналами зв'язку. Завдяки цьому, вся інформація рухається по колу в одному напрямку.

Робоча станція, яка приймає сигнали, розпізнає дані і отримує тільки ті повідомлення, які їй адресовані. У даній топології мережі застосовується маркерний доступ, що надає право на певний порядок використання кільця. Логічна топологія в даному випадку - логічне кільце.

Така мережа легко створюється і налаштовується. Єдиний недолік мережі топології кільце - якщо хоч в одному місці пошкоджена лінія зв'язку або вийшов з ладу, порушується працездатність всієї мережі.

Через деякій ненадійності, в чистому вигляді даний вид топології рідко застосовується. На практиці в основному застосовують модифікації різних кільцевих топологій.

Рис. 1.5 Топологія "Кільце"



Топологія мережі - **Token Ring**.

Така топологія ґрунтується на топології мережі «фізичне кільце із застосуванням типу зірка». Така топологія передбачає підключення всіх робочих станцій до центрального концентратора (або Token Ring), так само як при топології «фізична зірка». Таким чином, центральний концентратор за допомогою перемичок здійснює послідовне з'єднання виходів з одних станцій з входами інших станцій.

Концентратор забезпечує з'єднання кожної станції тільки з двома сусідніми станціями - попередньої і наступної. Робочі станції пов'язані між собою петлею кабелю, яка забезпечує передачу даних між станціями, тобто окрема станція ретранслює інформацію далі. Для забезпечення цього, кожна робоча станція

обладнана спеціальними приймально-передавальними пристроями, що дозволяють управління проходження даних в мережі.

Концентратор утворює основне первинне і резервне кільця. При обриві в основному кільці, його можна обійти, використовуючи для цього резервні кільця. Для цього застосовується чотирьохжильні кабель. У разі порушення роботи станції або при обриві лінії зв'язку мережа продовжує працювати, оскільки концентратор виключає несправну станцію, таким чином замикає кільце передачі даних.

Система Token Ring зроблена таким чином, що маркер передається по логічному кільцю між вузлами. Передача маркера має фіксований напрямок. Якщо станція має маркер, вона передає інформацію на наступну станцію.

Але для такої передачі даних робоча станції спочатку повинні дочекатися появи вільного маркера. Отриманий маркер містить всі адреси станції, яка направила цей маркер, в тому числі і станції, для якого він призначався. Наступна станція передає маркер далі по мережі, для наступної станції, і так далі по колу.

Головний вузол мережі (в основному це файл-сервер) маркер створює, далі цей маркер відправляється в мережу по кільцю. В даному випадку, такий вузол є активним монітором і строго стежить за рухом маркера, який не повинен загубитися або зруйнуватися.

До переваг такої топології Token Ring можна віднести:

- однаковий доступ до робочих станцій;
- надійність системи;
- стійкість до несправностей деяких станцій або при розривах з'єднань.

Недоліки Token Ring - це дуже велика витрата матеріалів на підключення, а відповідно, найдорожча розводка для ліній зв'язку.

1.2 Фізичні компоненти мереж

- Мережева карта
- Повторювач
- Концентратор(Hub)

- Комутатор
- Міст
- Маршрутизатор
- Міжмережевий екран

1.2.1 Концентратор (Hub)

Мережевий концентратор - пристрій для об'єднання комп'ютерів в мережу Ethernet із застосуванням кабельної інфраструктури типу звита пара. В даний час витіснені мережевими комутаторами.

Мережеві концентратори також могли мати роз'єми для підключення до існуючих мереж на базі товстого або тонкого коаксіального кабелю.

Концентратор працює на першому (фізичному) рівні мережевої моделі OSI, ретранслюючи вхідний сигнал з одного з портів в сигнал на всі інші (підключені) порти, реалізуючи, таким чином, властиву Ethernet топологію загальна шина, з поділом пропускної здатності мережі між усіма пристроями і роботою в режимі напівдуплекса. Колізії (тобто спроба двох і більше пристроїв почати передачу одночасно) обробляються аналогічно мережі Ethernet на інших носіях - пристрої самостійно припиняють передачу і відновлюють спробу через випадковий проміжок часу, кажучи сучасною мовою, концентратор об'єднує пристрої в одному домені колізій.

Мережевий концентратор також забезпечує безперебійну роботу мережі при відключенні пристрою від одного з портів або пошкодженні кабелю, на відміну, наприклад, від мережі на коаксіальному кабелі, яка в такому випадку припиняє роботу цілком.

1.2.2 Комутатор

Мережевий комутатор - пристрій, призначений для з'єднання декількох вузлів комп'ютерної мережі в межах одного або декількох сегментів мережі. Комутатор

працює на канальному (другому) рівні моделі OSI. Комутатори були розроблені з використанням мостових технологій і часто розглядаються як багатопортові мости. Для з'єднання декількох мереж на основі мережевого рівня служать маршрутизатори (3 рівень OSI).

На відміну від концентратора (1 рівень OSI), який поширює трафік від одного підключеного пристрою до всіх інших, комутатор передає дані лише безпосередньо отримувачу (виняток становить широкомовний трафік всіх вузлів мережі і трафік для пристроїв, для яких невідомий вихідний порт комутатора). Це підвищує продуктивність і безпеку мережі, позбавляючи інші сегменти мережі від необхідності (і можливості) обробляти дані, які їм не призначалися.

Комутатор зберігає в пам'яті (т.зв. асоціативної пам'яті) таблицю комутації, в якій вказується відповідність MAC-адреси вузла порту комутатора. При включенні комутатора ця таблиця порожня, і він працює в режимі навчання. В цьому режимі надходять на якийсь порт дані передаються на всі інші порти комутатора. При цьому комутатор аналізує фрейми (кадри) і, визначивши MAC-адресу хоста-відправника, заносить його в таблицю на деякий час. Згодом, якщо на один з портів комутатора надійде кадр, призначений для хоста, MAC-адреса якого вже є в таблиці, то цей кадр буде переданий тільки через порт, зазначений у таблиці. Якщо MAC-адреса хоста-одержувача не асоційована з яким-небудь портом комутатора, то кадр буде відправлений на всі порти, за винятком того порту, з якого він був отриманий. Згодом комутатор будує таблицю для всіх активних MAC-адрес, в результаті трафік локалізується.

Варто відзначити малу латентність (затримку) і високу швидкість пересилки на кожному порту інтерфейсу.

1.2.3 Маршрутизатор

Маршрутизатор - спеціалізований комп'ютер, який пересилає пакети між різними сегментами мережі на основі правил і таблиць маршрутизації. Маршрутизатор може пов'язувати різнорідні мережі різних архітектур. Для

прийняття рішень про пересилання пакетів використовується інформація про топологію мережі і певні правила, задані адміністратором.

Маршрутизатор працює на «мережевому» (третьому) рівні мережевий моделі OSI, на відміну від комутаторів і концентраторів (хабів), які працюють відповідно на другому і першому рівнях моделі OSI.

Зазвичай маршрутизатор використовує адресу одержувача, вказану в заголовку пакета, і визначає по таблиці маршрутизації шлях, по якому слід передати дані. Якщо в таблиці маршрутизації для адреси немає описаного маршруту - пакет відкидається.

Існують і інші способи визначення маршруту пересилки пакетів, коли, наприклад, використовується адреса відправника, використовувані протоколи верхніх рівнів і інша інформація, що міститься в заголовках пакетів мережевого рівня. Нерідко маршрутизатори можуть здійснювати трансляцію адреси відправника і одержувача, фільтрацію транзитного потоку даних на основі певних правил з метою обмеження доступу, шифрування / розшифрування даних, що передаються та інше.

1.3 Опис моделі OSI

Эталонна модель OSI - це описова схема мережі; її стандарти гарантують високу сумісність і здатність до взаємодії різних типів мережевих технологій. Крім того, вона ілюструє процес переміщення інформації по мережі. Це концептуальна структура, що визначає мережеві функції, реалізовані на кожному її рівні. Модель OSI описує, яким чином інформація проходить шлях через мережеве середовище (наприклад, дріт) від однієї прикладної програми (наприклад, програм обробки таблиць) до іншої прикладної програми, що знаходиться в іншому підключеному до мережі комп'ютері. По мірі того, інформація проходить вниз через рівні системи, вона стає все менше схожою на людську мову і все більш схожа на ту інформацію, яку розуміють комп'ютери, а саме на "одиниці" та "нулі". Эталонна модель OSI ділить задачу переміщення інформації між компьютерами через мережеву середу на

сім меленьких та недуже рівнів. Коженя із цих семи рівнів обраний тому, що вони відносно автономні, отже, вони легше вирішують без надмірної опори на зовнішню інформацію. Таке розділення на рівні називається ієрархічним представленням. Кожен рівень відповідає одній з підзадач

1.3.1 Рівні моделі OSI та їх функції

Кожний рівень модели OSI має спеціальні функції, що відповідають програмному забезпеченню.

1: Фізичний рівень

Фізичний рівень - це найнижчий рівень системи, який відповідає за кодування переданої інформації в рівень сигналів, прийнятий в середовищі передачі, і зворотнє декодування. Тут же визначаються вимоги до з'єднань, електричного узгодження, заземлення, захисту від завад.

2: Канальний рівень

Також називається рівень управління лінією передачі, відповідає за формування пакетів стандартного вигляду, що включають початкові та кінцеві керуючі поля. Тут проводиться управління доступом до мережі, виявляються помилки передачі і проводиться повторна передача примусових помилкових пакетів.

3: Мережевий рівень

Відповідає за адресацію пакетів та перекладі логічних імен в фізичні мережеві адреси (і назад), а також за вибір маршруту, за яким пакет доставляються за призначенням (якщо в мережі є кілька маршрутів)

4: Транспортний рівень

Сесійний рівень встановлює, управляє та розриває зв'язок між двома хостами. Цей рівень також синхронізує діалог між представницькими рівнями 2-х хостів і управляє їх обміном даних. Він розпізнає логічні імена абонентів, контролює надані їм права доступу.

5: Сеансовый рівень

Основна функція, що виконується на сеансовому рівні, нагадує роботу посередника або судді - управління діалогом між пристроями, що називаються також узлами. Взаємодія систем, організованих на цьому рівні, може відбуватися в трьох різних режимах: симплексному, полудуплексному і повнодуплексному. Сеансовий рівень зазвичай займається відділом даних однієї програми від інформації іншого додатка.

Нижче наведені деякі протоколи та інтерфейси сеансового рівня:

NFS (Network File System - мережева файлова система) Створено компанією Sun Microsystems і використовується на робочих станціях Unix разом з TCP / IP, щоб зробити доступ до віддалених ресурсів прозрачним для користувача.

SQL (Structured Query Language - мова структурированных запросів) У мові SQL, розробленому компанією IBM, користувач може в нелегкій формі визначити свої вимоги до інформації, доступ до якої здійснюється на локальних або віддалених системах.

RPC (Remote Procedure Call - визов віддалених процедур) Представляється простим інструментом переадресації в середовищі клієнт / сервер. Процедури RPC створюються на комп'ютері клієнта і виконуються на сервері.

X Window, Широко застосовується на інтелектуальних терміналах для зв'язку з віддаленими комп'ютерами Unix і дозволяє працювати з цими комп'ютерами, як з локальними.

ASP (AppleTalk Session Protocol - сеансовий протокол AppleTalk) Використовується в середовищі клієнт / сервер. Призначений для встановлення та підтримки сеансу між машинами клієнта і сервера по протоколу ASP.

DNA SCP (протокол керування сеансами цифрової мережі) - протокол сеансового рівня ДНК.

6: Представницький рівень

Представницький рівень, або рівень представлення даних, визначає, чи дійсні данні, що передаються прикладним рівнем однієї системи для використання

прикладным рівнем іншої системы, якщо ні - визначає та форматує данні. Тут вже виконується шифрування та дешифровка даних, а при необхідності - стискання.

7: Прикладний рівень

Прикладний рівень найбільш близький до користувача з усіх рівнів моделі OSI. Цей рівень надає мережеві сервіси користувача, такі як передача файлів, електронна пошта тощо. Рівень відрізняється від інших тим, що він не надає послуги іншими, тільки додатками в моделі OSI. Він також управляє іншими шести рівнями.

1.4 Види мережних атак

Мережі завжди чутливі до несанкціонованого моніторингу та різних типів мережевих атак. Якщо у вашій мережі не введено належних заходів безпеки та контролю, існує можливість для мережевих атак всередині та за межами вашої мережі.

1.4.1 DOS атаки

Ідея атаки DOS полягає в тому, щоб знизити якість послуг, пропонованих сервером, або аварійну зупинку серверу від великого навантаження. DoS (Denial of Service) атака не передбачає втручання в цільовий сервер. Як правило, це досягається шляхом перевантаження цільової мережі або цільового сервера або відправки мережевих пакетів, що може викликати надзвичайну плутанину в мережі або на сервері.

Атака "відмова в обслуговуванні" характеризується явною спробою злодіїв запобігти використанню цієї служби. Деякі з прикладів.

- Спроби "затопити" мережу, тим самим запобігаючи корисному мережевому трафіку.
- Спроби зривати з'єднання між двома машинами, тим самим запобігаючи доступ до сервісу.
- Спроби завадити певній особі отримати доступ до послуги.

- Спроби зривати роботу певній системі або особі.

Одна проста DoS (Denial of Service) атака була названа "Ping of Death". Ping of Death змогла використати простий інструмент для пошуку та усунення неполадок TCP / IP. Використовуючи інструмент ping, хакери можуть наповнити мережу великою кількістю пакетів, що може призвести до аварійного завершення цільового сервера.

1.4.2 SYN атака

Перш ніж розуміти, що таке атака SYN, ми повинні знати про тристоронній механізм рукостискання TCP / IP. Ініціюється сеанс протоколу керування передачею / Інтернет-протоколом (TCP / IP) за допомогою тристороннього рукостискання. Два комунікаційних комп'ютера обмінюються SYN, SYN / ACK та ACK, щоб розпочати сеанс. Ініціюючий комп'ютер надсилає пакет SYN, до якого приймаючий хост видає SYN / ACK та чекає відповіді ACK від ініціатора.

Напад атаки SYN - найпоширеніший тип атаки. Напад відбувається, коли зловмисник відправляє велику кількість пакетів SYN жертві, змушуючи їх чекати відповіді, які ніколи не прийдуть. Третя частина тристороннього рукостискання TCP ніколи не виконається. Оскільки хост чекає великої кількості відповідей, запити справжнього сервісу не обробляються. Адреса джерела цих SYN-пакетів в атаці SYN-повені зазвичай встановлюється як недоступний хост. У результаті неможливо знайти атакуючого комп'ютера.

Cookie SYN забезпечують захист від нападу SYN. SYN cookie реалізується за допомогою спеціального початкового номера TCP за допомогою програмного забезпечення TCP і використовується як захист від атак SYN Flood. Використовуючи дескриптивні брандмауери, які скидають очікувані з'єднання TCP після певного тайм-ауту, ми можемо зменшити ефект атаки SYN.

1.4.3 Сніффер атака

Сніффер - це програма, яка може захоплювати мережеві пакети. Sniffers також відомі як аналізатори мережевого трафіку. Хоча аналізатори протоколів - це справді інструменти для вирішення мережевих проблем, вони також використовуються хакерами для атак на мережі. Якщо мережеві пакети не шифруються, дані в пакеті мережі можуть бути прочитані за допомогою сніффера. Sniffing - це процес, який використовують нападники для захоплення мережевого трафіку за допомогою сніффера. Як тільки пакет захоплений за допомогою сніффера, вміст пакетів може бути проаналізований. Сніфери використовуються хакерами для захоплення чутливої інформації в мережі, такі як паролі, інформація про обліковий запис тощо.

Багато розвідників доступні для безкоштовного скачування. Провідні сніфери пакетів - це Wireshark, Dsniff, Etherpeek, sniffit тощо.

1.4.4 Man-in-the-middle атака

Атака "людина-по-середині" (MITM) - це тип атаки, де атакуючі втручаються в існуюче спілкування між двома комп'ютерами, а потім здійснюють моніторинг, захоплення та керування зв'язком. В атаці "людина-по-середині" втручання спрямоване на законних користувачей, щоб отримати контроль над мережевим зв'язком. Інший кінець шляху до зв'язку може повірити, що це ви і продовжуєте обмінюватися даними.

Атаки "людина в середині" (MITM) також відомі як "атаки викрадання сеансу", що означає, що атакуючий викрадає легітимний сеанс користувача для керування спілкуванням.

Багато профілактичних методів доступні для нападу "Людина-по-середині" (MITM), а деякі з них наведено нижче.

• Технології публічної інфраструктури ключових слів (PKI)

• Перевірка затримки спілкування

• Посилена взаємна автентифікація

1.4.5 Підробка IP адреси

Підробка IP-адреси - це тип нападу, коли зловмисник бере на себе джерело IP-адреси IP-пакетів, щоб він з'явився так, ніби пакет прийшов з іншої дійсної IP-адреси. Під час підробки IP-адреси IP-пакети створюються з підробленими IP-адресами джерела, щоб видати себе за інші системи або захистити ідентифікацію відправника.

Щоб чітко це пояснити, під час підробки IP-адреси інформація IP-адреси, розміщена у полі джерела заголовка IP-адреси, не є реальною IP-адресою вихідного комп'ютера, на якому було створено пакет. Змінюючи вихідну IP-адресу, фактичний відправник може зробити вигляд, що пакет був відправлений іншим комп'ютером, і отже відповідь з цільового комп'ютера буде відправлена на підроблену адресу, зазначену в пакеті, а також ідентифікація того, що зловмисник також захищений .

Пакетна фільтрація - це метод запобігання атакам підробки IP. Блокування пакетів з-за меж мережі з вихідною адресою всередині мережі (вхідна фільтрація) та блокування пакетів зсередини мережі з адресою джерела за межами мережі (фільтрація виходу) може допомогти запобігти атакам підробки IP-адрес.

1.4.6 ARP spoofing атака

Комп'ютер, підключений до локальної мережі IP / Ethernet, має дві адреси. Одина з них - MAC (Media Access Control), яка є глобально унікальною та незмінною адресою. MAC-адреси необхідні, щоб протокол Ethernet міг передавати дані назад і вперед, незалежно від того, які додаткові протоколи використовуються поверх неї. Ethernet надсилає та отримує дані на основі MAC-адрес. MAC-адреса також називається адресою Layer2, фізичною адресою чи адресою обладнання.

Інша адреса - це IP-адреса. IP - це протокол, який використовується програмами, незалежно від того, яка мережева технологія працює під нею. Кожен комп'ютер у мережі повинен мати унікальну IP-адресу для спілкування. Програми

використовують IP-адресу для спілкування. IP-адреса також називається адресою Layer 3 або Логічною адресою.

Щоб пояснити це більш чітко, програми використовують IP-адресу для спілкування та MAC-адресу низького рівня для зв'язку. Якщо для роботи програми на комп'ютері необхідно зв'язатися з іншим комп'ютером за допомогою IP-адреси, перший комп'ютер повинен дізнатись MAC-адресу другого комп'ютера, оскільки технології нижчого рівня Ethernet використовують MAC-адреси для передачі даних.

Операційні системи зберігають кеш ARP-відповідей, щоб мінімізувати кількість запитів ARP. ARP - протокол без статусу, і більшість операційних систем оновить кеш, якщо отримано відповідь, незалежно від того, чи вони відправили фактичний запит.

ARP (Address Resolution Protocol) Spoofing атака (ARP повінь або отруєння ARP) допомагає зловмисникові отримувати кадри даних по локальній мережі (LAN), та змінювати трафік і т.д. ARP Spoofing атаки проводяться шляхом відправки підроблених повідомлень ARP до локальної мережі Ethernet . Мета цього полягає в тому, щоб пов'язати MAC-адресу зловмисника з IP-адресою іншого комп'ютера, як правило, шлюзу за замовчуванням. Тут будь-який трафік, відправлений до шлюзу за замовчуванням, помилково надсилається атакуючому. Зловмисник може переслати трафік на фактичний шлюз за замовчуванням після того, як вилучити або змінити дані, перш ніж передати його.

1.4.7 Атака на DNS сервер

DNS - системи доменних імен. DNS є обов'язковим сервісом в мережах TCP / IP і перетворює доменні імена в IP-адреси. Комп'ютери в мережі спілкуються за допомогою IP-адреси. IP-адреси - це 32-бітні номери, які важко запам'ятати. Доменні імена бувають алфавітними і для людей їх легше запам'ятати. Коли ми використовуємо ім'я домену для зв'язку з іншим хостом, служба DNS повинна перевести ім'я на відповідну IP-адресу.

DNS-сервери зберігають базу даних доменних імен та відповідних IP-адрес. Атаки DNS Spoofing здійснюються шляхом зміни імені доменного імені легітимного сервера на сервері DNS, щоб вказати на якийсь IP, крім нього, а потім викрасти ідентифікацію сервера.

Взагалі існує два типи атак отруєння DNS; Отруєння DNS-кеш-пам'яті та виправлення DNS-ID.

У DNS-кеші отруєного DNS-серверу здійснюються хаписи, які не походять з авторитетних джерел доменних імен (DNS).

1.4.8 Фішінг та Фармінг атаки

Фішинг-атака підробки - це комбінація підробки електронної пошти та атаки зловмисників веб-сайтів. Фішинг-атакуючий починає фішинг-атаку, відправляючи масові електронні листи, видавши себе за веб-сайт, який вони підробляють. Як правило, електронні листи з метою фішингу, видаються за листи від законних фінансових організацій, таких як банки, попереджаючи користувача, що їм потрібно входити до свого облікового запису з тих чи інших причин. Посилання зазвичай надано в електронному листі на підроблений веб-сайт, що дуже схожий на веб-сайт банку. Коли користувач вводить комбінацію userid / password і відправляє ці данні, зловмисник викрадає ці значення, а веб-сторінка перенаправляється на реальний сайт.

Фармінг - це ще одна атака під тиском, в якій атакуючий намагається підробити DNS (Domain Name System), щоб трафік на веб-сайт таємно переадресовувався на фальшивий сайт, навіть якщо браузер, здається, відображає веб-адресу, яку ви хотіли відвідати.

Висновки до розділу 1

У першому розділі детально описано предмет дослідження, а саме види мережевих підключень, основні топології, види мережевого обладнання, а також основні види атак на локальні мережі та ресурси.

Можливі загрози будуть використанні у наступному розділі для побудови політики безпеки.

2 ФОРМАЛІЗАЦІЯ ПРАВИЛ ТА ПОЛІТИК БЕЗПЕКИ

В даному розділі обрано та описано усі правила та політики, що мають бути налаштовані на мережевому обладнанні для досягнення найліпших показників захищеності.

2.1 Перелік правил

Списки доступу

- Dynamic ARP inspection
- Port security
- Налаштування SSH
- EXEC timeout
- TACACS+ authenication
- Disable Unused Services
- Logging best practices
- Enhanced Crashinfo File Collection
- Network time protocol
- Login Password Retry Lockout
- Exclusive Configuration Change Access
- Cisco IOS Software Resilient Configuration

2.2 Опис поданих правил

- **Списки доступу**

Головний інструмент контролю трафіку. Коректне налаштування списків доступу.

Правило – кожен активний список доступу має бути “Білим”, та не повинен містити в одній строчці більше двох “any”.

Структура: (permit/deny)-(protocol)-(source ip)-(source port)-(destination ip)-(destination port)

Рядок доступу дозволяє або забороняє проходження трафіку заданого протоколу з певного порту хоста(або цілої мережі) до заданого хоста(або цілої мережі) за заданим портом. Правило забороняє написання рядків де явним чином використано 2 рази any. На рисунках нижче наведено приклади списків доступу, що задовольняють це правило та ні.

- **Dynamic ARP inspection**

Dynamic ARP inspection (DAI) може бути використана для пом'якшення атак ARP-отруєння на місцеві сегменти. DAI перехоплює та перевіряє зв'язок між IP та MAC-адресою всіх ARP-пакетів на ненадійних портах. У середовищах DHCP DAI використовує дані, створені за допомогою функції snooping DHCP. Пакети ARP, отримані на надійних інтерфейсах, не перевіряються, і недійсні пакети на ненадійних інтерфейсах відкидаються. У середовищах, відмінних від DHCP, використовується ACL.

Правило – мають бути активовані 2 команди:

```
!
ip dhcp snooping
ip dhcp snooping vlan <vlan-range>
!
```

та

```
!
ip arp inspection vlan <vlan-range>
!
```

- **Port security**

Port security використовується для контролю MAC-адреси в інтерфейсі доступу. Port Security може використовувати динамічно вивчені MAC-адреси, щоб полегшити початкову конфігурацію. Після того, як безпека порту визначила порушення MAC, вона може використовувати один із чотирьох режимів порушення. Ці режими - захищати, обмежувати, вимикати та вимикати VLAN. У випадках, коли

порт забезпечує доступ лише для однієї робочої станції з використанням стандартних протоколів, може бути достатньо максимальної кількості. Протоколи, які використовують віртуальні MAC-адреси, такі як HSRP, не працюють, якщо максимальне число встановлено на один.

Правило – на кожному активному інтерфейсі має бути увімкнена фільтрація або за однією MAC-адресою, або максимум за першими 3-ма:

```
!
interface <interface>
  switchport
  switchport mode access
  switchport port-security
  switchport port-security mac-address sticky
  switchport port-security maximum <number>
  switchport port-security violation <violation-mode>
!
```

- **Налаштування SSH**

SSH працює над надійним транспортним рівнем і забезпечує потужні можливості автентифікації та шифрування. Єдиним надійним транспортом, який визначено для SSH, є TCP. SSH забезпечує засоби безпечного доступу та виконання команд на іншому комп'ютері або пристрої через мережу. Функція протоколу безпечного копіювання (SCP), яка настроюється через SSH, дозволяє безпечно передавати файли.

Якщо команда ip ssh verson 2 не налаштована явно, тоді Cisco IOS підтримує SSH версії 1.99. SSH Version 1.99 дозволяє використовувати як SSHv1, так і SSHv2 з'єднання. SSHv1 вважається небезпечним і може мати несприятливий вплив на систему. Якщо SSH увімкнено, рекомендується вимкнути SSHv1 за допомогою команди ip ssh версії 2.

Правило – SSHv2 має бути увімкнений, таймаут не більше 120 секунд. Перевірити слід декілька команд:

```
!
ip ssh time-out 60
ip ssh authentication-retries 3
!
ip ssh version 2
!
line vty 0 4
transport input ssh
!
```

- **EXEC timeout**

Exec-timeout – вказує через який проміжок часу треба термінувати сессію, для запобігання доступу до обладнання 3 осіб. Команда exec-timeout має використовуватися на vty та tty рядках. За замовчуванням сесія завершається через десять хвилин бездіяльності. Часто системні адміністратори збільшують цей час для більшої зручностіу користуванні.

Правило – таймаут має бути меншим або рівним 9 хвилинам.

```
!
line con 0
exec-timeout <minutes> [seconds]
line vty 0 4
exec-timeout <minutes> [seconds]
```

- **TACACS+ authenication**

TACACS+ - це протокол автентифікації, який пристрої Cisco IOS можуть використовувати для аутентифікації користувачів на віддаленому сервері AAA. Ці користувачі можуть отримати доступ до пристрою за допомогою SSH, HTTPS, telnet або HTTP.

Аутентифікація TACACS+, або більш загальна автентифікація AAA, забезпечує можливість використання окремих облікових записів користувачів для кожного адміністратора мережі. Коли ви не залежите від єдиного спільного пароля, безпека мережі покращується, а ваша підзвітність посилюється.

Правило – на пристрої має бути налаштована автентифікація TACACS+:

```
!
aaa new-model
aaa authentication login default group tacacs+
!
tacacs-server host <ip-address-of-tacacs-server>
tacacs-server key <key>
!
```

- **Disable Unused Services**

Як найкраща практика безпеки, будь-яка непотрібна послуга повинна бути відключена. Ці непотрібні послуги, особливо ті, які використовують протокол обробки користувацьких даних (UDP), рідко використовуються для законних цілей, але можуть використовуватися для запуску служби доменів і інших атак, які перешкоджають фільтруванню пакетів.

Правило – наступні комани мають бути увімкнені:

```
!
no service tcp-small-servers
no service udp-small-servers
no ip finger
no ip bootp server ????
ip dhcp bootp ignore
no mop enabled ????
no ip domain-lookup
no service pad
no ip http server
no ip http secure-server
no service config
!
```

- **Logging best practices**

Журнал подій забезпечує контроль за роботою пристроїв Cisco IOS та мережі, в якій вони знаходяться. Програмне забезпечення Cisco IOS надає кілька гнучких параметрів реєстрації, які можуть допомогти досягти цілей організації та управління мережею.

Рекомендується надсилати інформацію про реєстрацію на сервер віддаленого робочого журналу. Це дає змогу ефективніше співвідносити та перевіряти події мережі на мережевих пристроях. Будь-які захисні засоби, які мережа надає трафіку керування (наприклад шифрування або зовнішній доступ), повинні бути розширені, щоб включити syslog трафік.

Правило – наступна команда має бути увімкнена:

```
!
logging host <ip-address>
!
```

- **Enhanced Crashinfo File Collection**

Функція «Розширені функції колекції файлів Crashinfo» автоматично видаляє старі файли з каталогом. Дозволяє пристрою видалити старі файли crashinfo, щоб створювати нові в разі збою системи. Ця функція також дозволяє конфігурувати кількість файлів crashinfo, які потрібно зберегти.

Правило – має бути увімкнена команда з кількістю файлв для зберігання 3.

```
!
exception crashinfo maximum files 3
!
```

- **Network time protocol**

Протокол мережевого часу (NTP) не є особливо небезпечною службою, але будь-яка непотрібна послуга може створювати вразливе місце. Якщо використовується NTP, важливо явним чином налаштувати надійне джерело часу та використовувати правильну автентифікацію. Точний і надійний час потрібний для системного журналу, наприклад, під час криміналістичних досліджень потенційних

нападів, а також для успішного з'єднання VPN, залежно від сертифікатів для аутентифікації фази 1.

Часовий пояс NTP - Коли ви налаштуєте NTP, часовий пояс потрібно налаштувати таким чином, щоб точні часові мітки були синхронізоані. Існує, як правило, два підходи до налаштування часового поясу для пристроїв у мережі з глобальною присутністю. Одним з методів є налаштування всіх мережевих пристроїв за допомогою уніфікованого часу (попередній час Гринвіча (GMT)). Інший підхід - налаштувати мережеві пристрої за допомогою локального часового поясу.

Правило – клієнт та сервер мають бути налаштовані наступним чином:

```
Client:
ntp authenticate
ntp authentication-key 5 md5 ciscotime
ntp trusted-key 5
ntp server 172.16.1.5 key 5
Server:
ntp authenticate
ntp authentication-key 5 md5 ciscotime
ntp trusted-key 5
!
```

- **Login Password Retry Lockout**

Функція блокування доступу до імені для входу в систему, дозволяє заблокувати локальний обліковий запис користувача після встановленої кількості невдалих спроб входу. Як тільки користувача заблоковано, його обліковий запис буде недоступний. Уповноважений користувач, якому надано рівень привілеїв 15, не може бути заблоковано за допомогою цієї функції. Кількість користувачів з рівнем привілеїв 15 має бути мінімальною.

Зверніть увагу, що авторизовані користувачі можуть блокувати себе, якщо буде досягнуто кількість спроб входу в невдалому режимі. Крім того, зловмисник

може створити умову відмови в наданні послуги (DoS), декілька раз повторюючи спробу автентифікації за допомогою дійсного ім'я користувача.

Правило – кількість невдалих спроб має дорівнювати 3

```
!
aaa new-model
aaa local authentication attempts max-fail <max-attempts>
aaa authentication login default local
!
username <name> secret <password>
!
```

- **Exclusive Configuration Change Access**

Функція ексклюзивної зміни доступу до конфігурацій забезпечує, що лише один адміністратор в певний момент часу може змінювати конфігурацію пристрою Cisco IOS. Ця функція допомагає усунути небажаний вплив одночасних змін, внесених до відповідних компонентів конфігурації. У автоматичному режимі конфігурація блокується сама, коли адміністратор видає команду configure EXEC-терміналу. У ручному режимі адміністратор використовує команду configure lock terminal, щоб заблокувати конфігурацію, коли вона входить до режиму налаштування

Правило – включена за замовчуванням ця функція.

```
!
configuration mode exclusive auto
!
```

- **Cisco IOS Software Resilient Configuration**

Ця функція дозволяє безпечно зберігати копію конфігурації пристрою Cisco IOS, яка в даний час використовується пристроєм Cisco IOS. Коли цю функцію ввімкнено, неможливо змінити або видалити файли резервної копії. Рекомендується ввімкнути цю функцію, щоб запобігти небажаним і шкідливим спробам видалення файлів бекапу.

Правило – наступні команди мають бути ввімкнуто.

```
!
secure boot-image
secure boot-config
!
```

2.3 Додаткові рекомендації

Усі зазначені правила можна розділити на 3 категорії, що забезпечують різні функціональні можливості, які потребують захисту.

- План керування
- Плану даних
- План контролю

План керування - керує трафіком, який надсилається на пристрій Cisco IOS, і складається з програм та протоколів, таких як Secure Shell (SSH) та Простий протокол керування мережею (SNMP).

План контролю - обробляє трафік, який є найважливішим для підтримки функціональності мережевої інфраструктури. План контролю складається з додатків і протоколів між мережевими пристроями, що включає протокол Border Gateway Protocol (BGP), а також Протоколи внутрішнього шлюзу (IGP), такі як Протокол маршрутизації внутрішнього шлюзу (EIGRP) та Open Shortest Path First (OSPF)

План даних - передає інформацію через мережевий пристрій. План даних не включає трафік, який надсилається на локальний пристрій Cisco IOS.

Безпечні операції

Захищена мережева робота є важливою темою. Хоча більшість цього доиплому присвячено безпечній конфігурації пристрою Cisco IOS, лише конфігурації не повністю захищають мережу. Операційні процедури, що використовуються в мережі, вносять стільки в безпеку, скільки в конфігурацію базових пристроїв.

Щоб зберегти безпечну мережу, нам слід знати про рекомендації Cisco, які були опубліковані. Нам необхідно знати про вразливість, перш ніж може бути оцінена загроза, яку вона може нанести мережі.

Використання аутентифікації, авторизації та обліку

Конфігурація аутентифікації, авторизації та обліку (AAA) необхідна для захисту мережевих пристроїв. Система AAA забезпечує аутентифікацію сеансів керування, а також може обмежувати користувачів конкретними командами, визначеними адміністратором, і записувати всі команди, введені всіма користувачами.

Централізація збирання журналів та моніторинг

Щоб отримати знання про існуючі, та історичні події, пов'язані з інцидентами безпеки, ваша організація повинна мати уніфіковану стратегію реєстрації подій та кореляції. Ця стратегія має залучати реєстрацію з усіх мережевих пристроїв і використовувати заздалегідь упаковані та настроювані можливості кореляції.

Після впровадження централізованого ведення журналу ви повинні розробити структурований підхід до аналізу журналів та відстеження інцидентів. Виходячи з потреб вашої організації, цей підхід може варіюватися від простого ретельного аналізу даних журналу до розширеного аналізу на основі правил.

Використовувати безпечні протоколи, коли це можливо

Багато протоколів використовуються для передачі чутливих даних керування мережею. Якщо це можливо, ви повинні використовувати безпечні протоколи. Вибір протоколу безпеки включає в себе використання SSH замість Telnet, щоб як дані автентифікації, так і інформація керування були зашифровані. Крім того, під час копіювання даних конфігурації необхідно використовувати захищені протоколи передачі файлів. Прикладом є використання протоколу безпечного копіювання (SCP) замість FTP або TFTP.

Збільшення видимості трафіку за допомогою NetFlow

NetFlow дозволяє відслідковувати потоки трафіку в мережі. Спочатку призначений для експорту інформації про дорожній рух до програм керування мережею, NetFlow також може використовуватися для показу інформації про потоки

маршрутизатора. Ця можливість дозволяє побачити, який трафік перетинає мережу в режимі реального часу. Незалежно від того, чи інформація про потоки експортується до віддаленого колектора, рекомендується настроїти мережеві пристрої для NetFlow, щоб вони могли бути використані реактивно, якщо це необхідно.

Конфігурація управління

Управління конфігурацією - це процес, за допомогою якого пропонуються, переглядаються, затверджуються та розгортаються зміни конфігурації. У контексті конфігурації пристрою Cisco IOS важливі два додаткових аспекти управління конфігурацією: архівація конфігурації та безпека.

Ви можете використовувати архіви конфігурації для відкату змін, внесених до мережевих пристроїв. У контексті безпеки архівні конфігурації також можуть бути використані для того, щоб визначити, які зміни безпеки були зроблені та коли відбулися ці зміни. У поєднанні з даними журнал AAA ця інформація може допомогти в перевірці безпеки мережевих пристроїв.

Конфігурація пристрою Cisco IOS містить багато чутливих деталей. Імена користувачів, паролі та вміст списків контролю доступу є прикладами цього типу інформації. Репозиторій, який ви використовуєте для архівування конфігурацій пристрою Cisco IOS, має бути захищено. Незабезпечений доступ до цієї інформації може підірвати безпеку всієї мережі.

Висновки до розділу 2

У третьому розділі було описано перелік правил, що бажано виконати для забезпечення найліпшого показника захищеності. У кожному підрозділі детально описано проблему, та які налаштування здатні її вирішити.

Ці правила будуть використані при написанні програми в наступному розділі.

3 СИСТЕМА АВТОМАТИЗОВАНОГО ЗБОРУ ТА ПЕРЕВІРКИ НАЛАШТУВАНЬ МЕРЕЖНОГО ОБЛАДНАННЯ

В розділі описується процес розробки та тестування програмного засобу, що дає змогу автоматизовано збирати та перевіряти налаштування мережевого обладнання.

3.1 Принцип роботи програми

Найкращий спосіб взаємодії з Cisco обладнанням є SSH інтерфейс. Для досягнення мети було обрано у якості інструменту мову програмування Python, та використано бібліотеку Paramiko для встановлення SSH з'єднання.

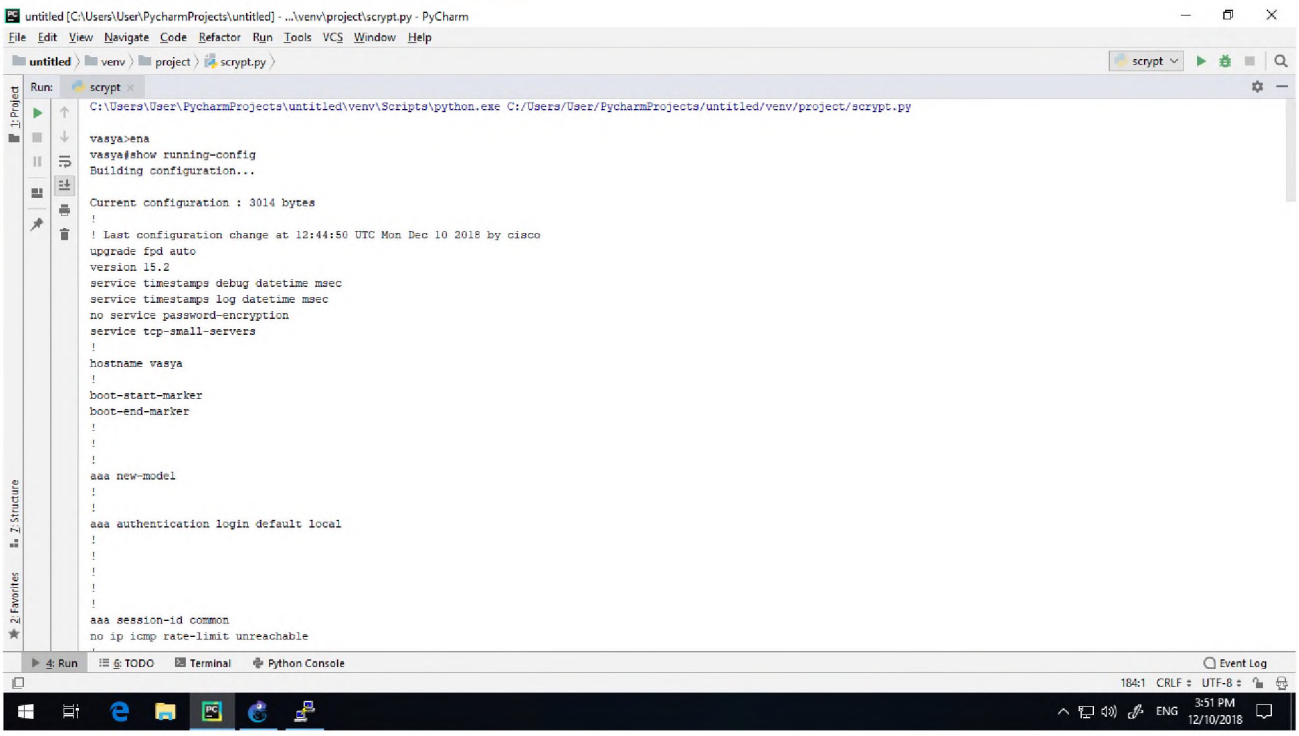
Принцип роботи програми:

1. Користувач вводить IP адресу, логін та пароль пристрою з яким потрібно встановити з'єднання

Рис.3.1 Форма для входу

2. Встановлюється SSH з'єднання
3. Програма отримує конфіги: "running-config" та розширений "running-config all"

Рис.3.2 Частина “running-config”



4. Робить перевірку активних списків доступу
5. У отриманих конфігах перевіряє інші правила наведені у розділі 3

3.2 Перевірка списків доступу

Ключовим етапом аудиту є перевірка активних списків доступу на наявність “any” у 2 полях.

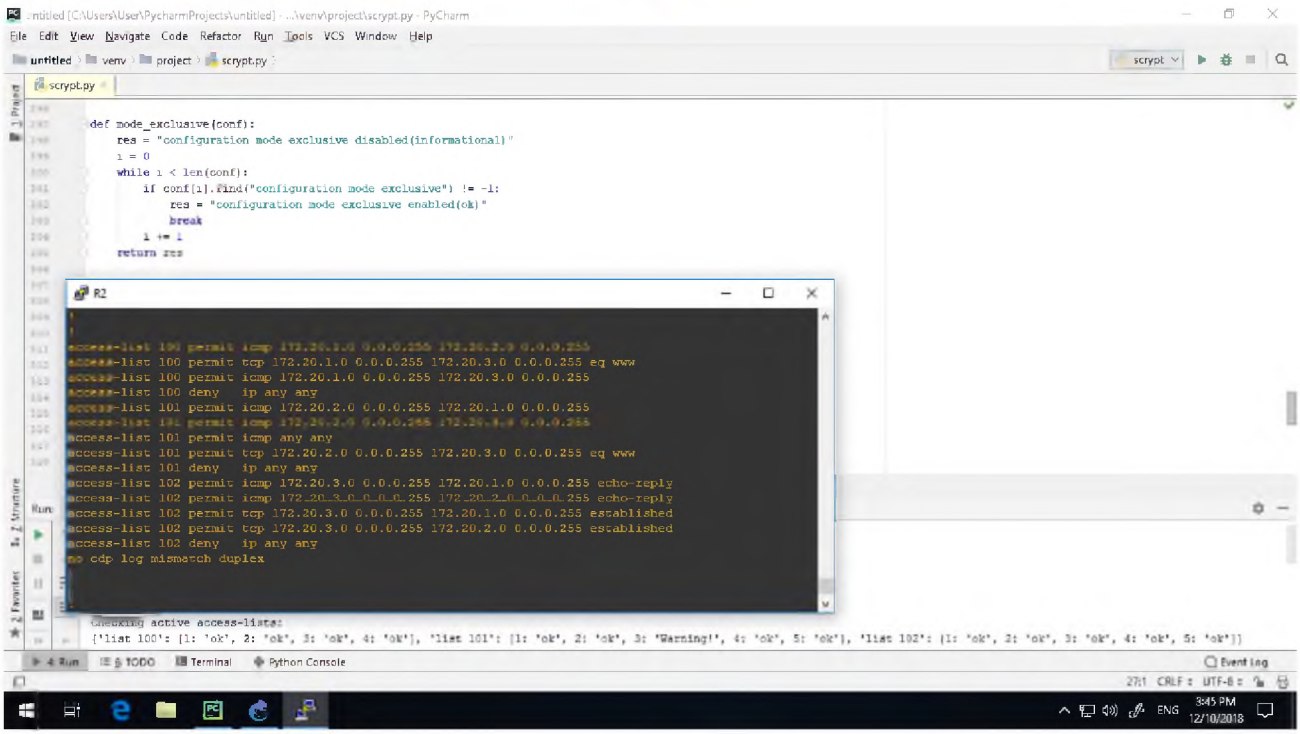
Є 3 результати, що може видати програма після перевірки одного рядка списку доступу:

- “Ok” – порушень не виявлено
- “Informational” – рядок може нести загрозу, аналітик має перевірити його особисто, спираючись на додаткову інформацію
- “Warning” – в даному рядку виявлена загроза, його обов’язково треба відредагувати

Для реалізації даного алгоритму було розроблено 4 функції:

1. Формує масив з номерами списків які активні на одному з інтерфейсів.
2. Отримує на вході масив, та для кожного елемента(списку доступу) створює масив строк. А також передає кожну строку у наступну функцію.
3. Отримує строку та проводить аналіз за розробленим принципом, потім повертає результат аналізу
4. По суті конструктор який використовує 3 функції зазначені вище, та формує зручний словник (базується на принципі “ключ->значення”) для виводу та перегляду результатів роботи

Рис.3.3 Результат аналізу списків доступу



У меншому вікні можна побачити існуючі списки доступу у кількості 3-х штук

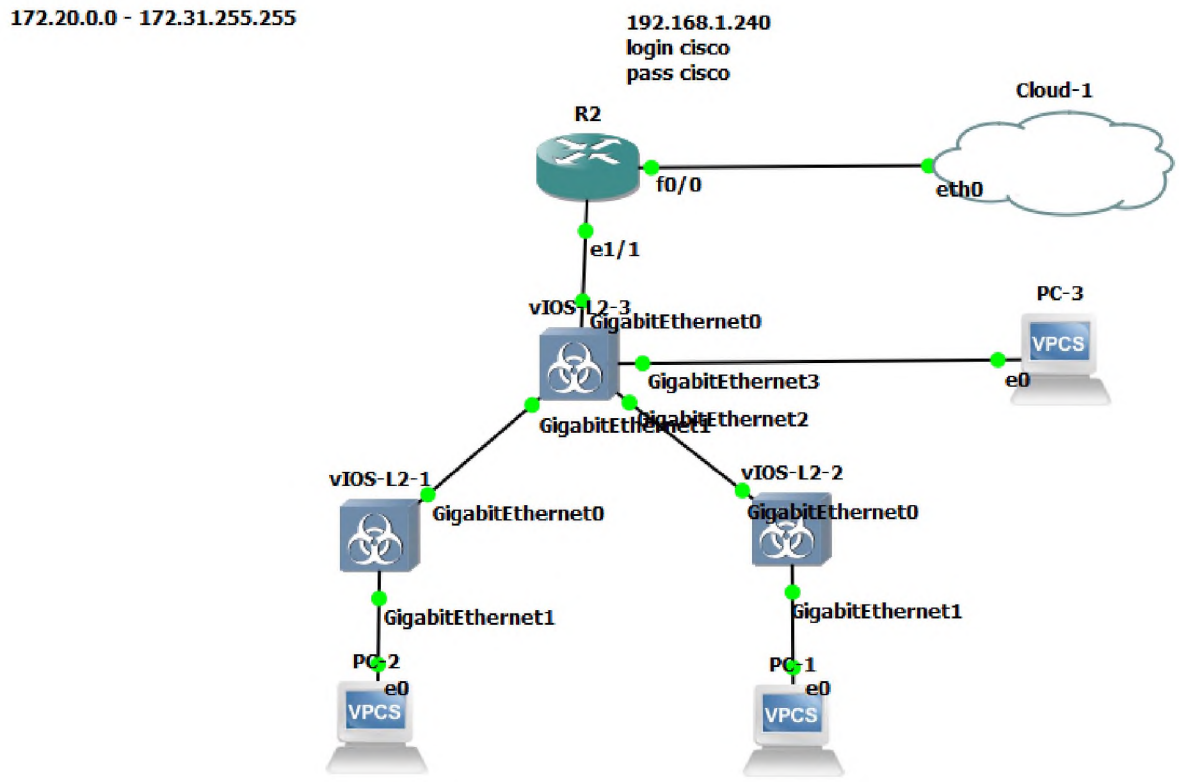
У виводі консолі Python результат перевірки кожної з строк за заданим правилом. По спискам доступу:

1. Усе добре
2. 3 рядок критична помилка
3. Усе добре

3.3Тестування розробленої програми

Для тестування було обрано застосунок GNS3, за допомогою якого стало можливим створити віртуальні маршрутизатори.

Рис. 3.3. Топологія мережі

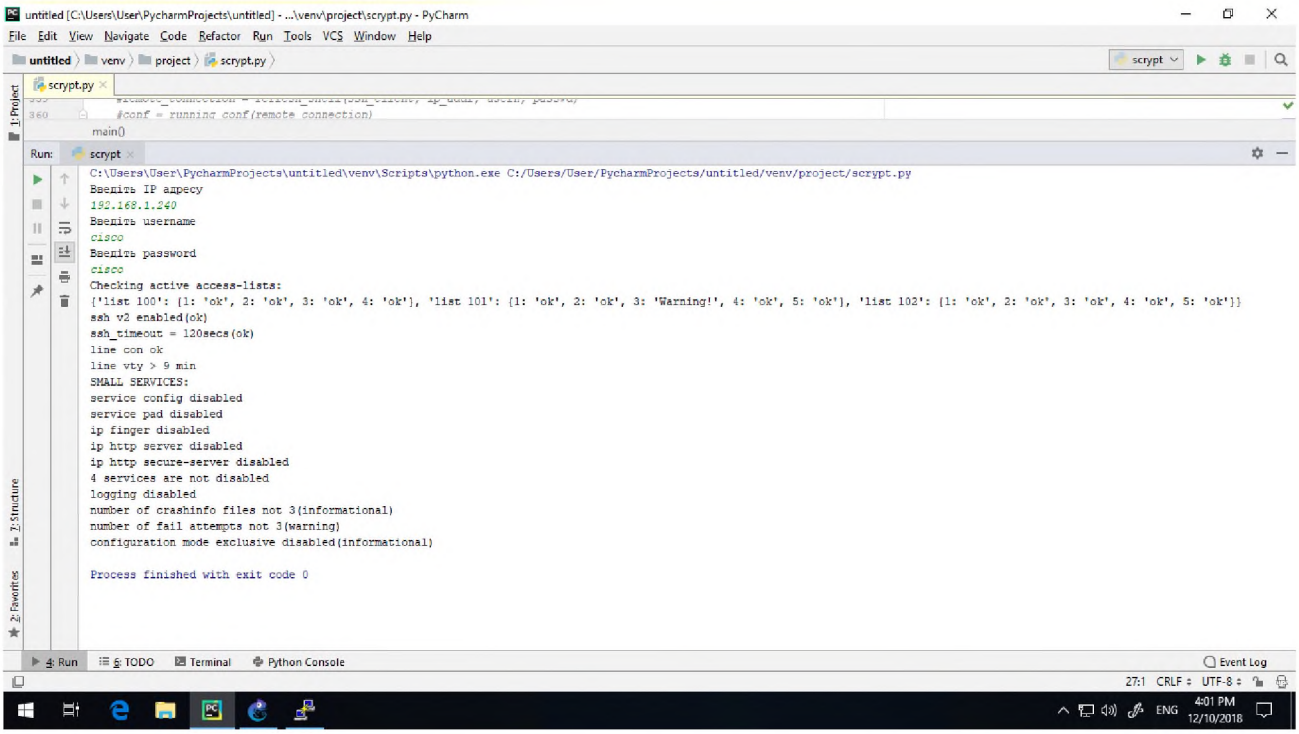


Топологія містить один маршрутизатор R2, а також 3 комутатори до яких підключені кінцеві користувачі. Комп'ютер PC-1знаходиться у VLAN100, PC-2 у VLAN200, PC-3 у VLAN300. Усі комутатори працюють у режимі L2.

Головним етапом аудиту такої мережі є якісна перевірка налаштувань саме маршрутизатора, бо саме на ньому термінується внутрішня мережа, та саме він маршрутизує пакети між вланами. На комутаторах проводяться лише деякі перевірки за необхідністю.

На рисунку 3.4 наведено результат роботи програми, щодо перевірки тестової мережі на відповідність заданим політикам.

Рис.3.4 Тестування програми



Висновки до розділу 3

У цьому розділі було розроблено функції, що реалізують перевірку заданих правил з розділу 2, а також програму в цілому. Використано застосунок GNS3 для розгортання тестової мережі на базі технологій від CISCO, а також протестовано в ньому розроблений програмний засіб.

Тестування проведено за 10 секунд, та можна вважати вдалим. Програма коректно реагує на усі зміни у файлах конфігів, та готова до застосування на справжньому обладнанні CISCO

4 СТАРТАП

В розділі необхідно провести маркетинговий аналіз перспектив реалізації запропонованого в роботі програмно-технічного рішення, а саме реалізації системи збору та аналізу мережевих налаштувань, а також оцінити можливість його ринкового впровадження.

4.1 Опис ідеї проекту

На першому кроці маркетингового аналізу необхідно описати ідею проекту. Послідовно потрібно проаналізувати: зміст ідеї (що пропонується), можливі напрямки застосування, основні вигоди, що може отримати користувач товару (за кожним напрямком застосування), чим відрізняється від існуючих аналогів та замінників.

Перші три пункти (табл. 5.1) дають цілісне уявлення про зміст ідеї та можливі базові потенційні ринки, в межах яких потрібно шукати групи потенційних клієнтів.

Таблиця 4.1- Опис ідеї стартап-проекту

Зміст ідеї	**Напрямки застосування**	**Вигоди для користувача**
Розробка програми, що буде автоматизовано перевіряти налаштування, та надавати короткий звіт.	Автоматизований аудит наявних мереж	Зменшення затрат людського часу на самостійну перевірку
	Створення нових мереж, та перевірка їх налаштувань	Отримуємо в кінці звіт, та одразу можно корегувати помилки

Аналіз потенційних техніко-економічних переваг ідеї передбачає визначення переліку техніко-економічних властивостей та характеристик ідеї, визначення

попереднього кола конкурентів, порівняльний аналіз показників (W - Слабка сторона, N - Нейтральна сторона, S - Сильна сторона).

Таблиця 4.2-Визначення сильних, слабких та нейтральних характеристик ідеї проекту

Техніко-економічні характеристики ідеї	**W** Слабка сторона	**N** Нейтральна сторона	**S** Сильна сторона
Економічні	Ринком збуту будуть скоріш за все компанії, що надають послугу аудиту мереж	Відсутність аналогів на ринку(не вдалось знайти)	Допомагає проводити аудит за декілька секунд, в результаті колосальна економія часу
Технологічні	Покриває лише мережі, що базуються на обладнанні Cisco		
Надійності			Усі рекомендації, що видає программа базуються на найкращих світових практиках

4.2 Технологічний аудит ідеї проекту

Далі проводиться аудит технології, за допомогою якої можна реалізувати ідею проекту (технології створення товару) (Таблиця 5.3).

Таблиця 4.3-Технологічна здійсненність ідеї проекту

	Ідея проекту	**Технології її реалізації**	**Наявність технологій**	**Доступність технологій**
	Використання SSH зеднання, як інтерфейсу взаємодії з Cisco обладнанням	Використання Python для алгоритмізації, та реалізації запланованих перевірок	Так	Усі технології доступні для використання
	Розробка власних політик бепеки, на основі найкращих практик	Аналіз практик, та синтез основних положень, щодо покращення загального стану захищенності	Так	Перелік найкращих практик в загальному доступі, його слід використовувати при побудові абсолютно усіх мереж на базі технології Cisco

Обрана технологія реалізації ідеї: використання можливостей Python, для налагодження SSH підключення, а також для алгоритмізації визначених правил. В такому випадку, технологічно реалізація ідей відбувається за допомогою дуже гнучкого механізму, що дає змогу до найменших деталей проаналізувати данні.

4.3 Аналіз ринкових можливостей запуску стартап-проекту

Визначення ринкових можливостей, які можна використати під час ринкового впровадження проекту, та ринкових загроз, які можуть перешкодити реалізації проекту, дозволяє спланувати напрями розвитку проекту із урахуванням стану ринкового середовища, потреб потенційних клієнтів та пропозицій проектів-конкурентів. Спочатку проводиться аналіз попиту: наявність попиту, обсяг, динаміка розвитку ринку (табл. 5.4).

Таблиця 4.4 Попередня характеристика потенційного ринку стартап-проекту

	Показники стану ринку (найменування)	**Характеристика**
	Кількість головних гравців, од	0 (попередньо)
	Загальний обсяг продаж, грн/ум.од	>10 мрлд. $
	Динаміка ринку (якісна оцінка)	Зростає
	Наявність обмежень для входу (вказати характер обмежень)	Обмежена кількість клієнтів, потрібна інтеграція з компаніями, що представляють такі послуги
	Специфічні вимоги до стандартизації та сертифікації	Залежить від законодавства окремої країни
	Середня норма рентабельності в галузі (або по ринку), %	>50%

За результатами аналізу таблиці можна сказати, що загалом ринок є досить привабливим, але існує значне обмеження – потенційними покупцями є компанії, що проводять аудит мереж.

Середня норма рентабельності в галузі (або по ринку) порівнюється із банківським відсотком на вкладення. За умови, що останній є вищим, можливо, має сенс вкласти кошти в інший проект. Але в даній галузі рентабельність є досить високою, а отже вхід в галузь більш привабливий, ніж вкладення коштів в банк.

Надалі визначаються потенційні групи клієнтів, їх характеристики, та формується орієнтовний перелік вимог до товару для кожної групи (табл. 5.5).

Таблиця 4.5 Характеристика потенційних клієнтів стартап-проекту

	Потреба, що формує ринок	**Цільова аудиторія (цільові сегменти ринку)**	**Відмінності у поведінці різних потенційних цільових груп клієнтів**	**Вимоги споживачів до товару**
	Значні витрати часу на не автоматизований аудит	Компанії, що займаються аудитом мереж	Компанія Аудитор, займається пошуком клієнтів за нас	Точність та швидкість перевірки
	Планується побудова власної мережі	Нова компанія, що будує власну мережу	Відмінностей майже немає. Окрім відсутності проміжної ланки(компанії Аудитора)	Точність та швидкість перевірки

Також окремо можна виділити підгрупи в відповідних сегментах. Компанії, що займаються аудитом або ті що планують запустити власну мережу можуть бути великими та середніми компаніями приватної та державної форми власності з відповідними особливостями функціонування.

Після визначення потенційних груп клієнтів проводиться аналіз ринкового середовища: складаються таблиці факторів, що сприяють ринковому впровадженню проекту, та факторів, що йому перешкоджають (табл. 5.6-5.7).

Таблиця 4.6 Фактори загроз

	Фактор	Зміст загрози	Можлива реакція компанії
	Покриває лише обладнання Cisco	Неможливо провести аудит мереж побудованих не на базі Cisco технологій	Розширювати кількість вендорів, що підтримує програмний продукт
	Відсутність імені продукту	Від імені на пряму залежить кількість продаж	Підвищення рейтингів, та постійний пошук компаній для інтеграцію продукту у їх повсякденну роботу

Таблиця 4.7 Фактори можливостей

	Фактор	Зміст можливості	Можлива реакція компанії
	Попит	У кожній компанії є мережі різних розмірів та складностей. Цим зумовлено попит на аудит мереж	Розширення ринків збуту

Надалі проводиться аналіз пропозиції: визначаються загальні риси конкуренції на ринку (табл. 4.8). Вказується тип конкуренції: монополія, олігополія, монополістична, чиста; рівень боротьби: локальний, національний, глобальний; галузева ознака: міжгалузева, внутрішньогалузева; вид: товарно-родова, товарно-видова, між бажаннями; характер переваг: цінова, не цінова; інтенсивність: марочна, не марочна. Описуються прояви відповідних характеристик та необхідні для конкурентоспроможності компанії.

Таблиця 4.8 Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	**В чому проявляється дана характеристика**	**Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)**
Тип конкуренції: чиста	Схожі програмні продукти знаходяться в рівних з нами обставинах	Постійний пошук нових компаній для інтеграції продукту
Рівень: глобальна	Ринок ІБ є глобальним, та по суті не жорстко географічно залежним	Використання англійської мови, як мови міжнародного спілкування, участь в міжнародних конференціях
Галузева ознака: внутрішньогалузева	Відбувається всередині галузі інформаційної безпеки	Розробка додаткових заходів з підвищення ефективності
За видами товарів: між бажаннями, товарно-родова	Конкуренція як в аспекті задоволення бажання побудувати процес чи покращити технологію, так і в засобах покращення технології	Розширення покриття бажань, що задовольняються, та/або вдосконалення існуючого методу
За характером конкурентних переваг: цінова та нецінова	Ціна важлива, але також важливі інші показники	Встановлення певного співвідношення ціна/вигода
За інтенсивністю: не марочна	Ціни можуть значно відрізнятися	Виділення унікальних рис своєї марки

На основі аналізу конкуренції, а також із урахуванням характеристик ідеї проекту (табл. 4.2), вимог споживачів до товару (табл. 4.5) та факторів маркетингового середовища (табл. 4.6-4.7) визначається та обґрунтовується перелік факторів конкурентоспроможності (табл. 4.9).

Таблиця 4.9 Обґрунтування факторів конкурентоспроможності

	Фактор конкурентоспроможності	**Обґрунтування (чинники, що роблять фактор значущим)**
	Ціновий	Вартість придбання послуги для клієнта досить важлива (для України зокрема, через придбання в форматі тендеру)
	Репутаційний	Значний вплив імені та репутації компанії, що надає послугу на вибір клієнта
	Технологічний	Як особливості технологічного покращення, так і співвідношення ціна/якість є дуже важливим аспектом у виборі конкретного товару споживачем

За визначеними факторами конкурентоспроможності (табл. 4.9) проводиться аналіз сильних та слабких сторін стартап-проекту (табл. 4.10).

Таблиця 4.10 Порівняльний аналіз сильних та слабких сторін проекту

№	**Фактор конкурентоспроможності**	**Бали 1-20**	**Рейтинг товарів-конкурентів (в порівнянні)**						
			3	**2**	**1**		**+1**	**+2**	**+3**
1	Ціновий	18						2	2
2	Репутаційний	12							
3	Технологічний	16						3	

Фінальним етапом ринкового аналізу можливостей впровадження проекту є складання SWOT-аналізу (матриці аналізу сильних (Strength) та слабких (Weak) сторін, загроз (Troubles) та можливостей (Opportunities) (табл. 5.11) на основі виділених ринкових загроз та можливостей, та сильних і слабких сторін (табл. 5.10).

Таблиця 4.11 SWOT-аналіз стартап-проекту

Сильні сторони:	**Слабкі сторони:**
простота використовуваного методу, новизна, мінімальні витрати на технологію	відсутність «імені», проблеми з початковими капіталовкладеннями
Можливості:	**Загрози:**
збільшення попиту на відповідний аудит, зростання видатків на ІБ департаменти	зменшення клієнтських бюджетів на ІБ, поява нових конкурентів

На основі SWOT-аналізу розробляються альтернативи ринкової поведінки (перелік заходів) для виведення стартап-проекту на ринок та орієнтовний оптимальний час їх ринкової реалізації з огляду на потенційні проекти конкурентів, що можуть бути виведені на ринок. Визначені альтернативи аналізуються з точки зору строків та ймовірності отримання ресурсів (табл. 5.12).

Таблиця 4.12 Альтернативи ринкового впровадження стартап-проекту

Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	**Ймовірність отримання ресурсів**	**Строки реалізації**
Участь в конкурсах стартапів, стартап-акселератори для отримання початкових вкладень	Відповідно до конкуренції в відборах	до року (місяці-роки)
Прямий пошук початкових інвестицій	Досить імовірно	місяці
Вихід з мінімальними початковими вкладеннями, еволюційний розвиток	≈ 1	Мінімально (тижні-місяці)

Загалом кожна з альтернатив може бути реалізованою, але з огляду на терміни реалізації доцільно спробувати отримати початкові інвестиції напряму (венчурні фонди, банківські кредити), а у випадку невдачі скористатись третьою альтернативою. Також можна паралельно брати участь у конкурсах.

4.4 Розроблення ринкової стратегії проекту

Розроблення ринкової стратегії першим кроком передбачає визначення стратегії охоплення ринку: опис цільових груп потенційних споживачів (табл. 4.13).

Таблиця 4.13 Вибір цільових груп потенційних споживачів

	Опис профілю цільової групи потенційних клієнтів	**Готовність споживачів сприйняти продукт**	**Орієнтовний попит в межах цільової групи (сегменту)**	**Інтенсивність конкуренції в сегменті**	**Простота входу у сегмент**
	Компанії, що займаються аудитом мереж	Можливо	Середній	між бажаннями	Середня
	Компанії, що будують свою мережу	Можливо	Середній	Між бажаннями	Середня

З огляду на загальну схожість характеристик обрано всі вказані цільові групи.

За результатами аналізу потенційних груп споживачів (сегментів) автори ідеї обирають цільові групи, для яких вони пропонуватимуть свій товар, та визначають стратегію охоплення ринку:

– якщо компанія зосереджується на одному сегменті – вона обирає стратегію концентрованого маркетингу;

– якщо працює із кількома сегментами, розробляючи для них окремо програми ринкового впливу – вона використовує стратегію диференційованого маркетингу;

– якщо компанія працює із всім ринком, пропонуючи стандартизовану програму (включно із характеристиками товару/послуги) – вона використовує масовий маркетинг.

Загалом на початковому етапі програма стандартизована, та робота відбувається в загальному по ринку, а отже застосовуватиметься масовий маркетинг. Але, сегменти мають певні відмінні один від одного риси, та бажано, принаймні в майбутньому використовувати стратегію диференційованого маркетингу.

Для роботи в обраних сегментах ринку необхідно сформувати базову стратегію розвитку (табл. 4.14).

Таблиця 4.14 Визначення базової стратегії розвитку

	Обрана альтернатива розвитку	**Стратегія охоплення ринку**	**Ключові конкурентоспроможні позиції відповідно до обраної альтернативи**	**Базова стратегія розвитку**
	Еволюційний розвиток з мінімальними початковими вкладеннями	Ексклюзивний розподіл	Низькі витрати, Індивідуальний підхід	Лідерство за витратами

Другий пункт ключових конкурентоспроможних позицій додає риси диференційованої стратегії розвитку, до лідерства за витратами. Адже індивідуальний підхід та наявність власних, не скопійованих особливостей «товару» притаманна саме цій стратегії.

Наступним кроком є вибір стратегії конкурентної поведінки (табл. 4.15).

Таблиця 4.15 Визначення базової стратегії конкурентної поведінки

Чи є проект «першопрохідцем» на ринку?	**Компанія шукатиме нових споживачів, або забиратиме існуючих у конкурентів?**	**Чи буде компанія копіювати основні характеристики товару конкурентів**	**Стратегія конкурентної поведінки**
Так	Переважно нових	Ні	Стратегія зайняття конкурентної ніші

На основі вимог споживачів з обраних сегментів до постачальника (стартап-компанії) та до продукту (табл. 4.5), а також в залежності від обраної базової стратегії розвитку (табл. 4.14) та стратегії конкурентної поведінки (табл. 4.15) розробляється стратегія позиціонування (табл. 4.16). що полягає у формуванні ринкової позиції (комплексу асоціацій), за яким споживачі мають ідентифікувати торгівельну марку/проект.

Таблиця 4.16 Визначення стратегії позиціонування

Вимоги до товару цільової аудиторії	**Базова стратегія розвитку**	**Ключові конкурентоспроможні позиції власного стартап-проекту**	**Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)**
Впровадження без порушення процесу функціонування,	Лідерство за витратами	Індивідуальний підхід, Мінімізація власних витрат	Виправдана вартість, Швидкий аудит,

помітний результат, коректне співвідношення ціни послуги та вигоди клієнта			Якісний аудит

4.5 Розроблення маркетингової програми стартап-проекту

Першим кроком є формування маркетингової концепції товару, який отримає споживач. Для цього (у табл. 4.17) потрібно підсумувати результати попереднього аналізу конкурентоспроможності товару.

Таблиця 4.17 Визначення ключових переваг концепції потенційного товару

	Потреба	**Вигода, яку пропонує товар**	**Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)**
	Зменшення затрат часу на проведення аудиту	Проведення аудиту стає в рази швидше	Аудит займає декілька секунд
	Структура аудиту, що відповідає найкращим практикам	Переформування структури з виділенням чітких правил	Індивідуальний підхід у визначенні правил та полтик

Надалі розробляється трирівнева маркетингова модель товару: уточнюється ідея продукту та/або послуги, його фізичні складові, особливості процесу його надання (табл. 4.18).

Таблиця 4.18 Опис трьох рівнів моделі товару

Рівні товару	**Сутність та складові**
І. Товар за	Спрощення процедури аудиту з використанням програмного

задумом	засобу, без зниження якості проведеного аудиту	
II. Товар у реальному виконанні	**Властивості/характеристики**	**Оцінка**
	Якість аудиту Зменшення затраченого часу на аудит	В кількісному вираженні оцінюється індивідуально
	Якість: вигода також оцінюється індивідуально, для підтвердження порівнюється результат з не автономним аудитом	
III. Товар із підкріпленням	До продажу: полегшення проведення аудиту та налаштування програми під кожного аудитора окремо	
	Після продажу: гарантійне обслуговування при виявленні помилок	
За рахунок чого потенційний товар буде захищено від копіювання: захист ідеї товару – унікальність застосовуваної реалізації		

Наступним кроком є визначення цінових меж, якими необхідно керуватись при встановленні ціни на потенційний товар (остаточне визначення ціни відбувається під час фінансово-економічного аналізу проекту), яке передбачає аналіз ціни на товари-аналоги або товари субститути, а також аналіз рівня доходів цільової групи споживачів. Аналіз проводиться експертним методом.

Наступним кроком є визначення оптимальної системи збуту, в межах якого приймається рішення (табл. 4.20):

– проводити збут власними силами або залучати сторонніх посередників (власна або залучена система збуту);

– вибір та обґрунтування оптимальної глибини каналу збуту;

– вибір та обґрунтування виду посередників.

Таблиця 4.20 Формування системи збуту

Специфіка закупівельної	**Функції збуту, які має виконувати**	**Глибина каналу збуту**	**Оптимальна система збуту**

поведінки цільових клієнтів	**постачальник товару**		
Тендерні закупівлі та пряме придбання послуги	Пошук клієнтів, установлення контактів Переговори Транспортування – Впровадження (в даному контексті)	Нульового рівня – виробник сам продає товар кінцевому споживачу	Збут власними силами – власна система

Останньою складової маркетингової програми є розроблення концепції маркетингових комунікацій, що спирається на попередньо обрану основу для позиціонування, визначену специфіку поведінки клієнтів (табл. 4.21).

Таблиця 4.21 Концепція маркетингових комунікацій

Специфіка поведінки цільових клієнтів	**Канали комунікацій, якими користуються цільові клієнти**	**Ключові позиції, обрані для позиціонування**	**Завдання рекламного повідомлення**	**Концепція рекламного звернення**
Закупівля через тендери, прямий пошук товару	Портали про публічні закупівлі, виставки, релізи, відгуки, прямі продажі	Задоволення специфічних потреб, реалізація певних переваг	Надати інформацію про товар (послугу) потенційним споживачам	Послуга допоможе вам зберегти ваш час, та поліпшити якість аудиту

В результаті аналізу вказаних таблиць описана ринкова (маркетингова) програма, що включає в себе концепції товару, збуту, просування та попередній аналіз можливостей ціноутворення, спирається на цінності та потреби потенційних

клієнтів, конкурентні переваги ідеї, стан та динаміку ринкового середовища, в межах якого буде впроваджено проект, та відповідну обрану альтернативу ринкової поведінки.

Висновки до розділу 5

У розділі проведено маркетинговий аналіз перспектив реалізації запропонованого в роботі науково-технічного рішення та оцінку можливості його ринкового впровадження.

Ринкова комерціалізація проекту є можливою, адже попит на ринку кібербезпеки взагалі зростає, та бюджети компаній на відповідні витрати збільшуються. Рентабельність надання описаного типу послуг є високою, з огляду на переваги. Швидкість проведення аудиту та якість самого аудиту.

Основною проблемою при впровадженні є відсутність так званого «імені» компанії, репутації, що є досить важливим аспектом у виборі постачальника послуг клієнтом. Водночас, кількість споживачів в цільових групах є обмеженою, а відновлюваність низька. Забезпечити відновлюваність можна за допомогою використання монетизації в форматі підписки, з тривалою підтримкою.

Імплементація проекту можлива, але є більш доцільною не в форматі окремого стартапу, а в якості підрозділу існуючої компанії.

ВИСНОВКИ

В результаті дипломної роботи мета, яка полягала в розробці системи автоматизованого збору та перевірки налаштувань мережевого обладнання може вважатися досягнутою.

Під час виконання роботи було визначено поняття та особливості функціонування мережевого обладнання та технологій пов'язаних з ним, наведено та проаналізовано можливі загрози, та сформульовано перелік правил, що базується на найкращих практиках налаштування мережевого обладнання Cisco. Розроблено програмний засіб, що власне проводить аудит, та видає звіт. А також протестовано його на віртуальному маршрутизаторі.

Отриманий програмний засіб дає змогу за 10 секунд проводити аудит одного маршрутизатора. Це допомагає скоротити згаяний час на декілька годин у порівнянні з ручною перевіркою усіх налаштувань.

Розроблений метод перевірки придатний для безпосереднього застосування при аудиті будь-яких мереж, що базуються на технологіях Cisco.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1 Cisco Guide to Harden Cisco IOS Devices [Електронний ресурс]:[Web-сайт]. – Режим доступу: https://www.cisco.com/c/en/us/support/docs/ip/access-lists/13608-21.html - 10.11.2018.

2 ISO 27001 [Електронний ресурс] - 10.11.2018.

3 Tallah Jarad: Mapping Cisco Security Solutions to ISO 27001 [Електронний ресурс]:- 10.11.2018.

4 Types of network attacks[Електронний ресурс]:[Web-сайт]. – Режим доступу: http://www.omnisecu.com/security/types-of-network-attacks.php - 22.04.2018.

5 Олексій Долгопольський ISSP Training Center: Network Defense Fundamentals [Електронний ресурс - 22.04.2018.

6 Лекция 31 Введение в безопасность сетей [Електронний ресурс]:[Web-сайт]. – Режим доступу: https://studfiles.net/preview/6449352/ - 22.04.2018.

ДОДАТОК А

```
import paramiko
import time

def start_test_lists(remote_connectrion):
    mass = active_list(remote_connectrion)
    i = 0
    res = {}
    while i < len(mass):
        res['list ' + mass[i]] = check_access_list(remote_connectrion, mass[i])
        i += 1
    return res

def check_access_list(remote_connection, name):
    remote_connection.send("show access-list " + name + "\n")
    time.sleep(1)
    dict = {}
    i = 0
    j = 0
    output = remote_connection.recv(52828)
    string = output.decode("utf-8")
    list = string.splitlines()
    while j < len(list) - 1:
        stri = list[0]
        if  len(stri) > 0 and stri.startswith("  ") != True:
            list.pop(0)
        if len(stri) == 0:
            list.pop(0)
        j += 1
```

```
    list.pop(len(list) - 1)
    while i < len(list):
        dict[i+1] = check_access_string(list[i])
        i+= 1
    return dict

def check_access_string(string):
    res = "ok"
    a = 10
    if "permit" in string:
        if "ip" in string:
            if string.count("any") == 1:
                res = "informational"
            if string.count("any") == 2:
                res = "Warning!"
        else:
            if string.count("any") >= 2:
                res = "Warning!"
            if string.count(".") <= 6 and res != "Warning!":
                try: last_index = string.rindex(".")
                except ValueError: a = 20
                if len(string) - last_index < 7:
                    res = "Warning!"
    return res

def active_list(remote_connection):
    remote_connection.send("show running-config | section ip access-group\n")
    time.sleep(3)
    output = remote_connection.recv(52828)
    string = output.decode("utf-8")
```

```
    list = string.splitlines()
    list.pop(len(list)-1)
    list.pop(0)
    list.pop(0)
    list.pop(0)
    i = 0
    line = """
    res = []
    while i < len(list):
        line = list[i]
        last_index = line.rindex(" ")
        res.append(line[17:last_index])
        i += 1
    return res

def ssh_version(mass):
    i = 0
    res = "ssh disabled(informational)"
    while i < len(mass):
        if mass[i].find("ip ssh version 1") != -1:
            res = "ssh v1 you must enable ssh v2(informational)"
            break
        if mass[i].find("ip ssh version 2") != -1:
            res = "ssh v2 enabled(ok)"
            break
        i += 1
    return res

def arp_inspection(mass):
    i = 0
```

```
    res = "arp inspection disabled"
    while i < len(mass):
        if mass[i].find("ip dhcp snooping") != -1:
            res = "dhcp snooping enabled"
            if mass[i].find("ip arp inspection") != -1:
                res = "arp inspection enabled"
        i += 1
    return res

def running_conf_all(remote_connection):
    #remote_connection.send("ena\n")
    remote_connection.send("show running-config all\n")
    i = 0
    time.sleep(5)
    while i < 30:
        remote_connection.send("    ")
        i += 1
        time.sleep(0.05)
    output = remote_connection.recv(100000)
    string = output.decode("utf-8")
    mass = string.splitlines()
    return mass

def running_conf(remote_connection):
    remote_connection.send("show running-config\n")
    i = 0
    time.sleep(5)
    while i < 5:
        remote_connection.send("    ")
        i += 1
```

```
        time.sleep(0.05)
    output = remote_connection.recv(52828)
    string = output.decode("utf-8")
    mass = string.splitlines()
    return mass

def ip_route(remote_connection):
    remote_connection.send("show ip route\n")
    time.sleep(1)
    output = remote_connection.recv(52828)
    string = output.decode("utf-8")
    return string

def ssh_timeout(remote_connection):
    remote_connection.send("show ip ssh | include Authentication timeout\n")
    time.sleep(2)
    i = 0
    output = remote_connection.recv(52828)
    string = output.decode("utf-8")
    mass = string.splitlines()
    check = """
    while i < len(mass):
        if mass[i].find("Authentication timeout:") != -1:
            check = mass[i]
            break
        i += 1
    last_index = check.rfind(" secs")
    res = "ssh_timeout = " + str(check[24:last_index]) + "secs"
    time1 = check[24:last_index]
    if int(time1) <= 120:
```

```
        res += "(ok)"
    else:
        res += "(warning)"
    return res

def line_vty(remote_connection):
    remote_connection.send("show running-config | section line vty\n")
    time.sleep(2)
    i = 0
    str = ""
    res = "line vty > 9 min"
    output = remote_connection.recv(52828)
    string = output.decode("utf-8")
    mass = string.splitlines()
    while i < len(mass):
        if mass[i].find("exec-timeout") != -1:
            str = mass[i]
            str = str[14:len(str)]
            pos = str.find(" ")
            str = str[0:pos]
            if int(str) <= 9:
                res = "line vty ok"
        i += 1
    return res

def line_con(remote_connection):
    remote_connection.send("show running-config | section line con\n")
    time.sleep(2)
    i = 0
    str = ""
```

```
    res = "line con > 9 min"
    output = remote_connection.recv(52828)
    string = output.decode("utf-8")
    mass = string.splitlines()
    while i < len(mass):
        if mass[i].find("exec-timeout") != -1:
            str = mass[i]
            str = str[14:len(str)]
            pos = str.find(" ")
            str = str[0:pos]
            if int(str) <= 9:
                res = "line con ok"
        i += 1
    return res

def small_services(conf_all):
    res = "SMALL SERVICES:\n"
    i = 0
    count = 0
    while i < len(conf_all):
        check = conf_all[i]
        if check.find("no service tcp-small-servers") != -1:
            res += "tcp-small-servers disabled\n"
            count += 1
            i += 1
            continue
        if check.find("no service udp-small-servers") != -1:
            res += "udp-small-servers disabled\n"
            count += 1
            i += 1
```

```
        continue
    if check.find("no ip finger") != -1:
        res += "ip finger disabled\n"
        count += 1
        i += 1
        continue
    if check.find("ip dhcp bootp ignore") != -1:
        res += "dhcp bootp ignore disabled\n"
        count += 1
        i += 1
        continue
    if check.find("no ip domain-lookup") != -1:
        res += "ip domain-lookup disabled\n"
        count += 1
        i += 1
        continue
    if check.find("no service pad") != -1 and res.find("service pad") == -1:
        res += "service pad disabled\n"
        count += 1
        i += 1
        continue
    if check.find("no ip http server") != -1:
        res += "ip http server disabled\n"
        count += 1
        i += 1
        continue
    if check.find("no ip http secure-server") != -1:
        res += "ip http secure-server disabled\n"
        count += 1
        i += 1
```

```
            continue
        if check.find("no service config") != -1:
            res += "service config disabled\n"
            count += 1
            i += 1
            continue
        i +=1
    res += str(9 - count) + " services are not disabled"
    return res

def logging(conf):
    res = "logging disabled"
    i = 0
    while i < len(conf):
        if conf[i].find("logging host") != -1:
            pos = conf[i].rfind(" ")
            ip = conf[i][pos+1:len(conf[i])]
            res = "logging enabled to host: " + ip
            break
        i += 1
    return res

def crashinfo(conf):
    res = "number of crashinfo files not 3(informational)"
    i = 0
    while i < len(conf):
        if conf[i].find("exception crashinfo maximum files 3") != -1:
            res = "number of crashinfo files 3 (ok)"
            break
        i += 1
```

```
    return res

def auth_fails(conf):
    res = "number of fail attempts not 3(warning)"
    i = 0
    while i < len(conf):
        if conf[i].find("aaa local authentication attempts max-fail 3") != -1:
            res = "number of fail attempts 3(ok)"
            break
        i += 1
    return res

def mode_exclusive(conf):
    res = "configuration mode exclusive disabled(informational)"
    i = 0
    while i < len(conf):
        if conf[i].find("configuration mode exclusive") != -1:
            res = "configuration mode exclusive enabled(ok)"
            break
        i += 1
    return res

def refresh_shell(ssh_client, ip_addr, usern, passwd):
    ssh_client.close()
    ssh_client.connect(hostname=ip_addr, username=usern, password=passwd)
    remote_connection = ssh_client.invoke_shell()
    remote_connection.send("ena\n")
    return remote_connection

def first_shell(ssh_client, ip_addr, usern, passwd):
```

```
    ssh_client.set_missing_host_key_policy(paramiko.AutoAddPolicy())
    ssh_client.connect(hostname=ip_addr, username=usern, password=passwd)
    remote_connection = ssh_client.invoke_shell()
    remote_connection.send("ena\n")
    return remote_connection

def test_router(ssh_client):
    print("Введіть IP адресу")
    ip_addr = input()
    print("Введіть username")
    usern = input()
    print("Введіть password")
    passwd = input()
    remote_connection = first_shell(ssh_client, ip_addr, usern, passwd)
    print("Checking active access-lists:")
    print(start_test_lists(remote_connection))
    remote_connection = refresh_shell(ssh_client, ip_addr, usern, passwd)
    conf = running_conf(remote_connection)
    remote_connection = refresh_shell(ssh_client, ip_addr, usern, passwd)
    conf_all = running_conf_all(remote_connection)
    remote_connection = refresh_shell(ssh_client, ip_addr, usern, passwd)
    print(ssh_version(conf))
    print(ssh_timeout(remote_connection))
    remote_connection = refresh_shell(ssh_client, ip_addr, usern, passwd)
    print(line_con(remote_connection))
    remote_connection = refresh_shell(ssh_client, ip_addr, usern, passwd)
    print(line_vty(remote_connection))
    print(small_services(conf_all))
    print(logging(conf))
    print(crashinfo(conf))
```

```
    print(auth_fails(conf))
    print(mode_exclusive(conf))
    return

def main():
    ip_addr = "192.168.1.240"
    usern = "cisco"
    passwd = "cisco"
    ssh_client = paramiko.SSHClient()
    remote_connection = first_shell(ssh_client, ip_addr, usern, passwd)
    #conf_all = running_conf_all(remote_connection)
    #remote_connection = refresh_shell(ssh_client, ip_addr, usern, passwd)
    conf = running_conf(remote_connection)
    print(conf)
    #test_router(ssh_client)
    ssh_client.close()

main()
```